

Structure-unknown non-linear dynamic systems: identification through neural networks

S F Masri†, A G Chassiakos‡ and T K Caughey§

†Department of Civil Engineering, University of Southern California, Los Angeles, California 90089, USA

‡School of Engineering, California State University, Long Beach, California 90840, USA

§Division of Engineering, California Institute of Technology, Pasadena, California 91125, USA

Received 24 November 1991, accepted for publication 6 December 1991

Abstract. This paper explores the potential of using parallel distributed processing (neural network) approaches to identify the internal forces of structure-unknown non-linear dynamic systems typically encountered in the field of applied mechanics. The relevant characteristics of neural networks, such as the processing elements, network topology, and learning algorithms, are discussed in the context of system identification. The analogy of the neural network procedure to a qualitatively similar non-parametric identification approach, which was previously developed by the authors for handling arbitrary non-linear systems, is discussed. The utility of the neural network approach is demonstrated by application to several illustrative problems.

1. Introduction

1.1. Background

The identification of mathematical models of physical structures on the basis of experimental measurements is a problem that has been receiving increasing attention in the recent past. The main impetus for this phenomenon is the central role that appropriate mathematical models have in active structural control applications. For example, the success of precision active control of large space structures is based on the availability of accurate mathematical models of these structures. Since the size, nature, and configuration of these large, flexible structures preclude their identification on the ground, future space missions have an essential need for efficient on-orbit system identification procedures. Detailed discussions of the many challenging problems encountered in the system identification of structural systems are available in the proceedings of some recent workshops convened to deal exclusively with model determination for large space systems [1].

Similarly, the potential for using active control approaches to reduce the response of large civil structures under arbitrary dynamic environments, such as earthquakes or wind, has been drawing a considerable amount of interest worldwide. In fact, the first 'US National Workshop on Structural Control Research' [2] was recently convened to address the research needs of

this rapidly emerging field. Among the key research topics identified by the attendees as needing much more investigation was the development of system identification approaches that can cope with the challenging nature of physical structures encountered in the structural mechanics field.

A related area in which there is also a rapidly increasing need for accurate mathematical models of large systems is the emerging field of 'health monitoring' of structures (civil as well as aerospace types). Techniques are being developed for damage assessment of large structures on the basis of system identification of high-fidelity mathematical models [3].

Numerous publications are available on the subject of system identification of structures [4]. A useful bibliography of some of the contemporary publications on the subject is available in the works of Beck [5] and Natke [6]. While much progress has been achieved in the general field of structural system identification, many challenging problems still await solution. The large modal analysis conferences [7] that are held annually attest to the wide ranging appeal of the subject to the structural engineering community and point out the level of research effort that is being devoted to resolving the main technical issues.

Even though there are many studies dealing with all computational aspects of system identification, there is a paucity of publications in the open literature that address

the use of parallel distributed processing methodologies (artificial neural networks) for identifying physical systems encountered in the applied mechanics field. It is this subject that the present paper explores.

1.2. Overview of the neural network literature

The pioneering work in this field is usually attributed to McCulloch and Pitts [8], who developed a simplified model of the neuron (the basic unit of the living brain) consisting of variable resistors and summing amplifiers. Over the intervening years seminal contributions were made by Hebb [9], Rosenblatt [10], Widrow [11], Minsky and Papert [12], Werbos [13], Albus [14], and Grossberg [15]. The interest in the analysis of neural networks was revived in the early 1980s due to the work of Hopfield [16], and later to the works of Hinton *et al* [17], Kohonen [18], McLelland and Rumelhart [19], and Mead [20]. Introductory background information for the non-specialist reader can be found in some recent publications such as Aleksander and Morton [21], Khanna [22], and Nelson and Illingworth [23].

In the recent explosive growth in the number of publications dealing with various aspects of neural nets, the numerous researchers investigating this topic recognize the fact that the field is still in its infancy. Much more work is needed in the future to exploit the full potential of neural nets for solving realistic problems.

Artificial neural network models have several inherent properties which distinguish them from traditional computational models [24].

- (i) Neural network models do not perform sequential computations as a Von Neuman machine does, but they are massively parallel nets of simpler computational elements. Analogue VLSI implementation of these networks provides much higher computation rates than sequential machines because of the inherent parallelism.
- (ii) Because neural networks contain many more processing elements than a sequential machine, and because these elements have primarily local connections, an artificial neural network has a higher degree of robustness or fault tolerance. If a few of the many links in the network are severed, the overall performance will not downgrade dramatically.
- (iii) Artificial neural networks are systems that have the property of adaptation or learning. Given a system and a set of 'training' patterns, the neural network can 'learn' about the system by adapting its internal parameters to match the training examples.

The above properties (i)–(iii) make artificial neural networks the ideal choice in cases in which real-time adaptation and fast processing of large amounts of data is required. For this reason a great deal of attention has been paid to artificial neural networks for high performance speech or image recognition.

1.3. Non-parametric identification of non-linear dynamic systems

The identification and modelling of non-linear dynamic systems through the use of experimental data is a problem of considerable importance in the applied mechanics area. Since the model structure in many practical dynamics problems is by no means clear, an increasing amount of attention has recently been devoted to non-parametric identification methods. These methods do not identify the physical parameters of the system (such as mass, stiffness, etc), but instead identify the parameters of a mathematical model which fits the input/output data.

One rather general non-parametric non-linear identification approach is based on the expansion of the non-linear restoring force functions in a power series or generalized Fourier series involving orthogonal polynomial functions. In applications, it is generally assumed that such series are rapidly convergent so that only a few terms of the series need be retained for identification purposes. In such an approach, the coefficients of the retained terms from the series become parameters of the system which may be identified by many well-known techniques, such as least-squares fit in the time domain.

The origins of this basic approach are classical and diverse, with roots in the theory of analytic functions and in the theory of Fourier series, and with applications in many engineering disciplines as well as operations research, economics, and the physical sciences. With regard to the engineering literature, the basic approach is outlined in the book by Graupe [25]. Applications of the method in the mechanical sciences appear to have originated in the early 1950s in several NACA technical notes [26–30], and in the papers by Klotter [31] and Shinbrot [32]. In the following years, interest in similar time series methods for non-linear system identification of structures expanded, as attested to by the representative publications of Kohr [33], Hoberock and Kohr [34], Sprague and Kohr [35], Sehitoğlu and Klein [36], Masri and Caughey [37], Masri *et al* [38], Natke [6], Shinozuka *et al* [39], Masri *et al* [40], Tomlinson [41], Hac and Spanos [42], Crawley and Aubert [43], Crawley and O'Donnell [44], Masri *et al* [45, 46], and Masri *et al* [47].

As a motivation for the present study, reference is made to the authors' paper [37] in which a simple non-parametric identification approach was introduced. The main idea behind this approach is that, for non-linear systems typically encountered in the applied mechanics field, the 'restoring force' acting on the system can be conveniently expressed in terms of suitable basis functions associated with the structure-unknown system. For example, in the case of a single-degree-of-freedom (SDOF) system, the restoring force can be expressed as a function of the system displacement and velocity (or other generalized coordinates):

$$g(y, \dot{y}) = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} C_{ij} T_i(y) T_j(\dot{y}) \quad (1)$$

where the C_{ij} are a set of undetermined constants and $T_i(\cdot)$ are suitable basis functions, such as orthogonal

polynomials, and y' , \dot{y}' are transformations of the generalized coordinates as in (2)

$$y' = [y - (y_{\max} + y_{\min})/2]/[(y_{\max} - y_{\min})/2]. \quad (2)$$

The range of summation indices n_1 and n_2 depends on the nature of the system and the desired degree of accuracy in estimating g . While there is a wide choice of suitable basis functions for least-squares applications, the orthogonal nature of the Chebyshev polynomials and their 'equal-ripple' characteristics make them convenient to use in conjunction with non-linear dynamic systems.

If the Chebyshev polynomials, given by

$$T_n(\xi) = \cos(n \cos^{-1} \xi) \quad -1 < \xi < 1 \quad (3)$$

and satisfying the weighted orthogonality property

$$\int_{-1}^1 w(\xi) T_n(\xi) T_m(\xi) d\xi = \begin{cases} 0, & n \neq m \\ \pi/2, & n = m \neq 0 \\ \pi, & n = m = 0 \end{cases}$$

where $w(x) = (1 - x^2)^{-1/2}$ is the weighting function, are used, then the C_{kl} coefficients would be given by

$$C_{kl} = \begin{cases} (2/\pi)^2 S_{kl}, & k \text{ and } l \neq 0 \\ (2/\pi^2) S_{kl}, & k \text{ or } l = 0 \\ (1/\pi^2) S_{kl}, & k \text{ and } l = 0 \end{cases}$$

where

$$S_{kl} = \int_0^\pi \int_0^\pi g(\cos^{-1} v_1, \cos^{-1} v_2) T_k(\theta_1) T_l(\theta_2) d\theta_1 d\theta_2 \quad (4)$$

and

$$v_i = \cos \theta_i \quad i = 1, 2. \quad (5)$$

Note that in the special case where no cross-product terms (in the generalized coordinates) are involved in function g , it can be expressed as the sum of two one-dimensional orthogonal polynomial series instead of a single two-dimensional series of the type under discussion.

The preceding steps can be geometrically interpreted as constructing a surface in three-dimensional space which characterizes the resistance force of the structure. The authors have previously demonstrated the validity and robustness of this approach through several analytical and experimental studies dealing with a variety of non-linear phenomena (polynomial non-linearities, hysteresis, limited slip) occurring in diverse physical systems ranging from an actual three-story building that underwent substantial non-linear deformations under earthquake ground motion [48], to a laboratory study of an aerospace bearing assembly incorporating Coulomb friction and limited slip non-linearities [47]. Applications of this technique to non-linear structures incorporating joints are reported in the work of Masri *et al* [49, 50], Crawley and Aubert [43], and Crawley and O'Donnell [44].

1.4. Scope

The purpose of this paper is to introduce a new non-

parametric method for the identification of structure-unknown non-linear dynamic systems. The method is based on the use of artificial neural networks as system identifiers.

In particular, we are interested in the determination of the system structural stiffness forces and inherent damping characteristics (which constitute the 'restoring forces' or internal forces). It is assumed that the mass distribution of structural components can be directly measured or accurately estimated.

Sections 2.1–2.6. of this paper present some pertinent background information from the neural network field. In presenting the necessary background, this paper also serves as a convenient, compact tutorial of some of the cogent material needed by researchers in the applied mechanics field who may wish to use neural nets for other computational mechanics tasks. Sections 3.1–3.3. present the development of an identification procedure for structural systems by neural nets, as well as applications to structural and non-linear dynamical systems.

2. Formulation

2.1. Artificial neural network models

An Artificial Neural Network (ANN) is a system with inputs and outputs, composed of a number of similar non-linear processing elements. These processing elements operate in parallel and are arranged in patterns similar to the patterns found in biological neural nets. The processing elements or nodes are connected to each others by adjustable weights [24, 51]. Changing these weights will change the input/output behaviour of the network, hence the following is a natural goal for such a system: *choose the weights of the net in such a way as to achieve a desired input/output relationship*. To achieve this goal, systematic ways of adjusting the weights have to be developed, which are named by the general term 'training' or 'learning' algorithms. A neural net is characterized by the following.

- (i) The processing elements.
- (ii) The network topology.
- (iii) The learning algorithm.

Each of these network characteristics is described next.

2.2. The processing elements (nodes)

A typical node sums n weighted inputs u_1, u_2, \dots, u_n and a bias term b and passes the result through a non-linear function $\gamma(\cdot)$ (figure 1).

$$v = \gamma(\bar{v}) \quad \bar{v} = \sum_{i=1}^n w_i u_i + b \quad (6)$$

Note that the bias term b can be considered either as

- (1) an input $u_0 = b$ connected to the node by a fixed weight $w_0 = 1$; or
- (2) a constant input $u_0 = 1$ connected to the node by a weight $w_0 = b$. In this case the value of b is

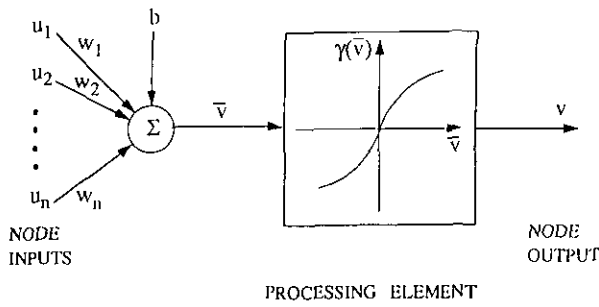


Figure 1. A typical processing element (node).

adjusted by the same algorithm as the rest of the weights in the net.

Typical non-linearities used in the nodes are

hard-limiters

$$\gamma(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases} \quad (7)$$

threshold operations

$$\gamma(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (8)$$

the logistic function (sigmoid)

$$\gamma(x) = \frac{1}{1 + e^{-\alpha x}} \quad \alpha > 0 \quad (9)$$

the modified hyperbolic tangent function

$$\gamma(x) = \frac{1 - e^{-\alpha x}}{1 + e^{-\alpha x}} \quad \alpha > 0. \quad (10)$$

The choice of the non-linearity depends on the particular application for which the network is being used. In cases when it is required that the non-linearity is differentiable, the logistic and the hyperbolic tangent functions are widely used. Moreover, their derivatives are given by

$$\gamma'(x) = -\alpha \gamma(x)[\gamma(x) - 1] \quad (11)$$

when $\gamma(x)$ is the logistic function, and by

$$\gamma'(x) = -(\alpha/2)[\gamma(x) + 1][\gamma(x) - 1] \quad (12)$$

if $\gamma(x)$ is the modified hyperbolic tangent.

2.3. The network topology

The network topology depends on the way the nodes are connected to each other and to the input and output vectors. According to their topology, neural networks can be classified as:

- single layer networks, when only one layer of nodes is present, namely the output layer (figure 2), or
- multi-layer networks, when the nodes are arranged in more than one layer (figure 3).

Moreover, a network can be characterized as

- a feedforward network, if there is no feedback to

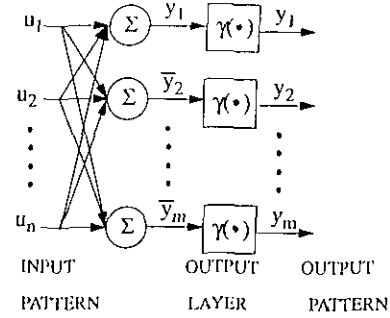


Figure 2. A single layer neural network.

previous layers from the output of subsequent layers (figure 3); or

- a recurrent network, if such a feedback connection exists.

2.4. The training algorithms—back propagation

By the term training or learning algorithm we mean a systematic procedure for adjusting the weights in the network to achieve a desired input/output relationship. A general feature of 'supervised' learning algorithms is that an error criterion $E = f(y, y_d)$ is evaluated, where y is the actual output vector and y_d is the desired output vector. The weights w_{ij} of the net are then adjusted in such a way as to reduce the value of the error criterion. The various learning algorithms reported in the literature differ in the ways this adjustment is being done. As an example, the Hebbian rule for a single layer net computes the weight change Δw_{ij} as being proportional to the product of the node outputs

$$\Delta w_{ij} = k v_i v_j \quad (13)$$

where v_i and v_j are the outputs of nodes i and j .

Another example is the celebrated 'Back Propagation' training algorithm, which is a gradient descent procedure based on a square error criterion. The back propagation algorithm is the one used for the applications demonstrated in sections 3.1–3.2. and is described in detail next, for the three-layered net of figure 3. The following notation is used (as in Narendra and Parthasarathy, [52]):

$\mathbf{u} = [u_1, u_2, \dots, u_n]^T = n \times 1$ input vector

$\mathbf{v} = [v_1, v_2, \dots, v_p]^T = p \times 1$ vector, output of layer 1 (first hidden layer)

$\mathbf{z} = [z_1, z_2, \dots, z_q]^T = q \times 1$ vector, output of layer 2 (second hidden layer)

$\mathbf{y} = [y_1, y_2, \dots, y_m]^T = m \times 1$ vector, output of the net

$\tilde{\mathbf{v}} = [\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_p]^T = p \times 1$ vector, intermediate output of layer 1 (first hidden layer)

$\tilde{\mathbf{z}} = [\tilde{z}_1, \tilde{z}_2, \dots, \tilde{z}_q]^T = q \times 1$ vector, intermediate output of layer 2 (second hidden layer)

$\tilde{\mathbf{y}} = [\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_m]^T = m \times 1$ vector, intermediate output of layer 3 (output layer)

$\mathbf{W}^i = \{w_{ij}^i\} = p \times n$ matrix of weights connecting the j th input to the i th node of layer 1

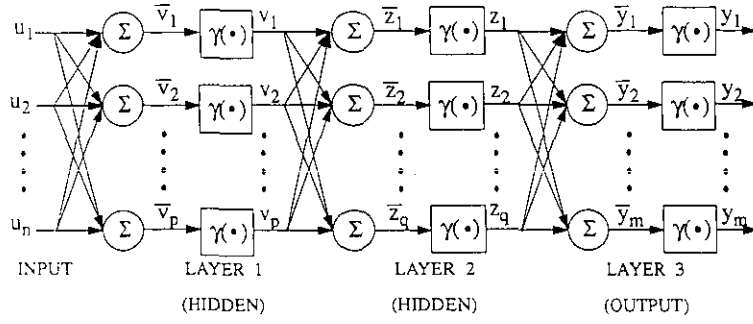


Figure 3. A feedforward three-layer neural network.

$\mathbf{W}^2 = \{w_{ij}^2\} = q \times p$ matrix of weights connecting the j th node of layer 1 to the i th node of layer 2

$\mathbf{W}^3 = \{w_{ij}^3\} = m \times q$ matrix of weights connecting the j th node of layer 2 to the i th node of layer 3

$\gamma(x)$ is the differentiable non-linearity $\gamma(x) = (1 - e^{-\alpha x}) / (1 + e^{-\alpha x})$; $\alpha > 0$

The various inputs and outputs in each layer are related by the following relations:

Layer 1:

$$\bar{\mathbf{v}} = \mathbf{W}^1 \mathbf{u} \quad v_i = \gamma(\bar{v}_i) \quad i = 1, \dots, p$$

Layer 2:

$$\bar{\mathbf{z}} = \mathbf{W}^2 \mathbf{v} \quad z_i = \gamma(\bar{z}_i) \quad i = 1, \dots, q$$

Layer 3:

$$\bar{\mathbf{y}} = \mathbf{W}^3 \mathbf{z} \quad y_i = \gamma(\bar{y}_i) \quad i = 1, \dots, m.$$

For clarity of presentation and notational uniformity, the bias terms b_i for each node have been incorporated into the adjustable weights as explained in section 2.2.; consequently, they do not appear explicitly in the network diagram.

In the case of 'supervised' learning, the network is being presented as a large number of pairs of input vectors and desired output vectors ($\mathbf{u}^r, \mathbf{y}_d^r$), where the superscript r ranges over all pairs used to train the network and the subscript d stands for the 'desired' output vector. During training, the network learns to associate the input vector \mathbf{u}^r with the output vector \mathbf{y}_d^r . For a given set of weights, if the network is presented an input \mathbf{u}^r it will produce an output \mathbf{y}^r , which should be identical or very close to \mathbf{y}_d^r if the training is successful.

In the back propagation algorithm, the performance criterion to be minimized by an appropriate choice of the weights is defined as

$$E = \frac{1}{2} \sum_r \sum_k (y_k^r - y_{dk}^r)^2 \quad (14)$$

where the index r ranges over all training patterns and the subscript k refers to the k th component of the \mathbf{y}^r and \mathbf{y}_d^r vectors.

The performance criterion is a measure of the distance between the output vectors \mathbf{y}^r and the desired outputs \mathbf{y}_d^r . The criterion E is a function of the actual output values y_k^r , which in turn depend on the values of the network weights $\{w_{ij}\}$ and the input patterns.

Because of this dependence of y_k^r on w_{ij} , the criterion E can be reduced if the weights w_{ij} are adjusted appropriately. In order to find a set of parameters (weights) that will reduce E , we need to calculate the gradient of E with respect to the parameters $\{w_{ij}\}$ and then adjust the parameter set in the direction of the negative gradient. The back propagation algorithm facilitates the calculation of this gradient, starting at the output layer and 'propagating' the results backwards to the first layer.

Let us assume that the network is to be trained over only one pattern, hence $r = 1$. Let

$$e_k = y_k - y_{kd} \quad k = 1, \dots, m \quad (15)$$

be the k th component of the output error. The performance criterion is then written as

$$E = \frac{1}{2} \sum_{k=1}^m e_k^2 \quad (16)$$

Now recall that

$$y_k = \gamma(\bar{y}_k) \quad \text{and} \quad \bar{y}_k = \sum_{j=1}^q w_{ks}^3 z_s \quad (17)$$

and consequently the derivative of y_k , with respect to a typical element w_{ij}^3 of matrix \mathbf{W}^3 , will be

$$\frac{\partial y_k}{\partial w_{ij}^3} = \begin{cases} 0, & \text{if } i \neq k \\ \gamma'(\bar{y}_k) z_j, & \text{if } i = k \end{cases} \quad (18)$$

from which it follows that

$$\frac{\partial e_k^2}{\partial w_{ij}^3} = \begin{cases} 0, & \text{if } i \neq k \\ 2e_k \gamma'(\bar{y}_k) z_j, & \text{if } i = k. \end{cases} \quad (19)$$

The derivative of the error criterion with respect to a typical element w_{ij}^3 of matrix \mathbf{W}^3 will then be

$$\frac{\partial E}{\partial w_{ij}^3} = e_i \gamma'(\bar{y}_i) z_j = \delta_i^3 z_j \quad (20)$$

where

$$\delta_i^3 = e_i \gamma'(\bar{y}_i). \quad (21)$$

The $(m \times q)$ gradient matrix of the error criterion with respect to the elements w_{ij}^3 of matrix \mathbf{W}^3 is now given as

$$\nabla_3 E = \left\{ \frac{\partial E}{\partial w_{ij}^3} \right\} = \begin{pmatrix} \delta_1^3 z_1 & \dots & \delta_1^3 z_q \\ \vdots & & \vdots \\ \delta_m^3 z_1 & \dots & \delta_m^3 z_q \end{pmatrix}. \quad (22)$$

If the elements of matrix \mathbf{W}^3 are adjusted in the direction of the negative gradient matrix, the performance criterion E will be reduced.

The dependence of the error e_k on a typical element w_{ij}^2 of matrix \mathbf{W}^2 is more involved, since the elements w_{ij}^2 do not appear explicitly in the calculation of e_k . From $y_k = \gamma(\bar{y}_k)$ we get

$$\frac{\partial y_k}{\partial w_{ij}^2} = \gamma'(\bar{y}_k) \frac{\partial \bar{y}_k}{\partial w_{ij}^2} \quad (23)$$

and from the definition of \bar{y}_k and the application of implicit differentiation it follows that:

$$\frac{\partial y_k}{\partial w_{ij}^2} = \gamma'(\bar{y}_k) \left(\sum_{s=1}^q w_{ks}^3 \gamma'(\bar{z}_s) \frac{\partial \bar{z}_s}{\partial w_{ij}^2} \right). \quad (24)$$

From

$$\bar{z}_s = \sum_{i=1}^p w_{si}^2 v_i \quad (25)$$

we get

$$\frac{\partial \bar{z}_s}{\partial w_{ij}^2} = \begin{cases} 0, & \text{if } i \neq s \\ v_j, & \text{if } i = s \end{cases} \quad (26)$$

hence

$$\frac{\partial y_k}{\partial w_{ij}^2} = \gamma'(\bar{y}_k) w_{ki}^3 \gamma'(\bar{z}_i) v_j. \quad (27)$$

From this it follows that

$$\frac{\partial e_k^2}{\partial w_{ij}^2} = 2e_k \gamma'(\bar{y}_k) w_{ki}^3 \gamma'(\bar{z}_i) v_j \quad (28)$$

and the derivative of the error criterion E with respect to a typical element w_{ij}^2 of matrix \mathbf{W}^2 , is given as

$$\frac{\partial E}{\partial w_{ij}^2} = \gamma'(\bar{z}_i) v_j \sum_{k=1}^m w_{ki}^3 \delta_k^3 = \delta_i^2 v_j \quad (29)$$

where

$$\delta_i^2 = \gamma'(\bar{z}_i) \sum_{k=1}^m w_{ki}^3 \delta_k^3. \quad (30)$$

The $(q \times p)$ gradient matrix of the error criterion, with respect to elements w_{ij}^2 of matrix \mathbf{W}^2 , is given as

$$\nabla_2 E = \left\{ \frac{\partial E}{\partial w_{ij}^2} \right\} = \begin{pmatrix} \delta_1^2 v_1 & \dots & \delta_1^2 v_p \\ \vdots & & \vdots \\ \delta_q^2 v_1 & \dots & \delta_q^2 v_p \end{pmatrix}. \quad (31)$$

A comparison of the two derivatives $\partial E / \partial w_{ij}^3 = \delta_i^3 z_j$ and $\partial E / \partial w_{ij}^2 = \delta_i^2 v_j$ shows that they are of the same form: a multiplication of the output of the previous layer (z_j or v_j) by some calculated signal (δ_i^3 or δ_i^2). Their difference lies in the way the signals δ_i^3 and δ_i^2 are calculated, depending on whether we are referring to the output layer or to a hidden layer. It is noted that in order to compute δ_i^2 , the signal δ_i^3 (which depends on the output layer) should be computed first. Thus, the term 'back propagation' is justified by the fact that signal δ_i^3 is propagated back to the previous layer.

Now the derivative of the error criterion with respect

to a typical element w_{ij}^1 of matrix \mathbf{W}^1 is easily seen to be

$$\frac{\partial E}{\partial w_{ij}^1} = u_j \gamma'(\bar{v}_i) \sum_{k=1}^q w_{ki}^2 \delta_k^2 = \delta_i^1 u_j \quad (32)$$

where

$$\delta_i^1 = \gamma'(\bar{v}_i) \sum_{k=1}^q w_{ki}^2 \delta_k^2. \quad (33)$$

The $(p \times n)$ gradient matrix of E with respect to elements w_{ij}^1 of matrix \mathbf{W}^1 is given as

$$\nabla_1 E = \left\{ \frac{\partial E}{\partial w_{ij}^1} \right\} = \begin{pmatrix} \delta_1^1 u_1 & \dots & \delta_1^1 u_n \\ \vdots & & \vdots \\ \delta_p^1 u_1 & \dots & \delta_p^1 u_n \end{pmatrix} \quad (34)$$

which has the same form as $\nabla_3 E$ and $\nabla_2 E$.

After the gradient matrices have been computed, the weights are adjusted in the negative gradient direction, so that the error criterion E is reduced. A typical weight w_{ij} (which could belong to any layer) is adjusted from its old value w_{ij}^{old} to its new value w_{ij}^{new} according to

$$w_{ij}^{\text{new}} = w_{ij}^{\text{old}} - \eta \partial E / \partial w_{ij}^{\text{old}}. \quad (35)$$

The stepwise η is termed the 'learning rate'. This η is usually chosen as constant during training, but it could also be adjustable, if this facilitates the training process.

2.5. Neural-network based identification of dynamical systems

Although the identification of dynamical systems is a subject that has attracted researchers for many years, it is only very recently that artificial neural networks have been used for this purpose. The non-linear mappings between inputs and outputs of a neural network and the ability of the net to adapt its parameters so as to minimize an error criterion, make the use of artificial neural networks particularly well suited for the identification of both non-linear and linear dynamical systems. During the past two years a few papers have appeared in the literature dealing with system identification using neural networks. In the work of Chu *et al* [53], a parametric identification of a system is performed. A least squares estimation procedure is implemented using a Hopfield optimization net. In the work of Kraft and Campagna [54], a neural network is used in conjunction with a classical controller for the control of dynamical systems. The net is trained in a manner similar to that of a CMAC, (cerebellar model articulation controller, reported in [14]). In the work of Lan [55], a neural network is used to 'learn' the inverse dynamics of an unknown system. The procedure is applied to a linear second order system. Some authors [56, 57] have pointed out the relation between the neural network identification approach and that of using the well-known radial basis function expansions for the identification of non-linear systems. It is shown that the radial basis function technique is a special case of a two layered neural network. In the work of Narendra and Parthasarathy [52], the notion of 'generalized' neural networks is introduced: generalized neural networks are nets connected to linear systems and to other

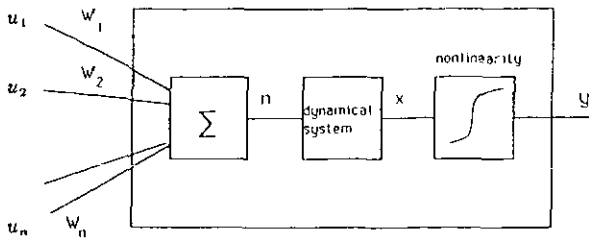


Figure 4. A dynamic neuron.

neural networks through some basic cascade and/or feed-back configurations. In the paper, generalized neural network configurations are used for the identification and control of discrete-time linear and non-linear systems.

2.6. Neural networks with dynamic neurons

The neural networks described in sections 2.1–2.4 are nets with static non-linearities in the nodes, since the function $\gamma(x)$ is a static function of its inputs (no dynamics involved). These networks work very well in modelling linear or non-linear static functions, but they do not perform well in modelling dynamic systems. Experience with neural network architectures has shown that good modelling of systems with dynamics requires the introduction of some dynamics into the network. Different architectures and various training algorithms have been proposed in the literature for the solution of this problem. Narendra and Parthasarathy [58] used ‘generalized’ back-propagation algorithms for training the ‘generalized’ neural networks in which the dynamics are lumped into a dynamical linear system, which in turn is connected to a static neural net, through cascade or feedback connections. Werbos [59] used a form of the back propagation algorithm called ‘back propagation through time’ for training of networks that model dynamic systems. Polycarpou and Ioannou [60] employed architectures that enabled them to provide stability and convergence proofs for the networks using Lyapunov function techniques. Chassiakos *et al* [61] and Kosmatopoulos *et al* [62] propose an architecture that gives rise to neural networks with dynamic neurons: each neuron contains a (not necessarily linear) dynamic system in addition to the non-linear sigmoid function (figure 4). The neuron output y is related to the neuron inputs u_1, \dots, u_N by the following equations [63, 64]:

$$n = \sum_{j=1}^N w_{ij}^k u_j \quad \dot{x} = -a(x)[b(x) + n] \quad y = \gamma(x) \quad (36)$$

where x is the neuron dynamic state. The network is trained by a gradient technique termed ‘distributed dynamic back propagation’. The results obtained suggest that superior performance can be achieved by such an architecture, which is based on distributing the dynamics over the network nodes rather than lumping them into one dynamical system as is done in the ‘generalized neural networks’ case.

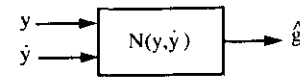


Figure 5. Block diagram for a neural identifier of a structural system.

3. Applications

3.1. Identification of structural systems

So far, very little work has been reported in the literature concerning the identification of structural systems by neural networks. In this section, the problem of identifying non-linear structural systems by neural networks is formulated and the identification procedure is developed.

Specifically, let the multi-degree of freedom (MDOF) system to be identified obey the following equation of motion:

$$M\ddot{y}(t) + g(y(t), \dot{y}(t)) = u(t) \quad (37)$$

where y is the system displacement vector, $g(y, \dot{y})$ is the unknown vector of restoring forces, M is the known mass matrix and $u(t)$ is the system excitation. This formulation represents general classes of structural systems, such as: systems with linear force/deflection characteristics, non-linear systems having polynomial-form non-linearities (such as the Duffing oscillator), as well as hysteretic systems. Hysteretic systems have a non-linearity that not only involves cross-product terms of displacement and velocity, but is not even expressible in polynomial form. Such systems, widely encountered in all areas of applied mechanics (e.g. building and equipment systems, as well as aerospace structures containing collapsible or retractable elements), are among the more difficult types of non-linear properties to investigate and identify [65–73].

The neural identifier is shown in block diagram form in figure 5. The network, represented by N , is of undetermined size, topology and node non-linearities, since these characteristics are application dependent. Presently, no concrete analytical results exist and no specific guidelines are available concerning the criteria for the selection of the network characteristics. The selection of network size, topology and node characteristics is based largely on experience and extensive experimentation with different networks.

The inputs to the net are the displacement and velocity vectors y, \dot{y} . The output of the net is a non-linear transformation of the inputs: $N(y, \dot{y})$. This output $N(y, \dot{y}) = \hat{g}$ should approximate the unknown function $g(y, \dot{y})$ of equation (37) if the network is a good identifier of the system. Note a slight change of notation between this section and section 2.4.: here, y and \dot{y} are the network inputs (represented by u in section 2.4.), \hat{g} is the network output (represented by y in section 2.4.), and u is the system excitation.

Assume that the experimental measurements for $u(t)$ and $\ddot{y}(t)$ are available and that the corresponding system displacement $y(t)$ and velocity $\dot{y}(t)$ vectors can be found by direct measurements or through integration of $\ddot{y}(t)$. If

the measurements are taken at discrete times t_k :

$$y_k = y(t_k) \quad \dot{y}_k = \dot{y}(t_k) \quad \ddot{y}_k = \ddot{y}(t_k) \quad u_k = u(t_k)$$

the values $g_k = g(y(t_k), \dot{y}(t_k))$ can then be computed from

$$g_k = u_k - M\ddot{y}_k.$$

During the training phase, the network is presented with the sequence of input vectors $\{[y_k, \dot{y}_k]^T\}$ and the sequence $\{g_k\}$ of desired output vectors. Given a set of weights (which is initially chosen randomly), the input vector $\{[y_k, \dot{y}_k]^T\}$ is propagated forward through the net at time t_k and the network output \hat{g}_k is calculated. This \hat{g}_k is compared to the desired output g_k . The performance criterion is then evaluated and its gradient $\partial E / \partial \{w_{ij}\}$ is computed with respect to the network parameters. The weights are now adjusted according to (35), and the procedure is repeated until an acceptable value of the error criterion is obtained. Now, given an input vector $[y_k, \dot{y}_k]^T$, the network will produce an output \hat{g}_k close to the desired output g_k . At this stage, the neural net has 'learned' the system.

During the validation phase, the network is given other input vector sequences $\{[y_\alpha, \dot{y}_\alpha]^T\}$ not among those used for training. If the training was successful and the network is a good identifier, it should produce an output sequence $\{\hat{g}_\alpha\}$ very close to the actual system output $\{g_\alpha\} = \{g(y_\alpha, \dot{y}_\alpha)\}$.

Specific applications on identification of the internal forces of linear structural systems by neural networks are presented in the works of Chassiakos and Masri [74, 75]. The use of multilayer networks with static nodes to identify the internal forces in some typical non-linear structural systems under a variety of deterministic as well as stochastic excitations is demonstrated in the work of Masri *et al* [76]. A comparison of the neural approach to non-parametric identification techniques (which are based on series expansion through Chebyshev polynomials), shows that the two approaches, although theoretically different, have remarkable qualitative similarities.

The authors' non-parametric identification method, which does not require any information concerning the nature of the underlying non-linearity:

- (1) determines the optimum coefficients of a doubly-indexed series in order to construct analytically a surface in three-dimensional space whose topography approximates the data set used for identification, and
- (2) uses interpolation techniques to estimate what the restoring force should be when given a new (i.e. not from the data set used for identification) set of coordinates in the state-space of the system.

On the other hand, if the task of system identification through the use of neural nets is viewed as a curve fitting of the data in a high-dimensional space, then the main concepts in neural nets ('learning' and 'generalization') are seen to be very similar to the above mentioned steps: starting with a complex data set obtained from a structure-unknown system, the neural net procedure:

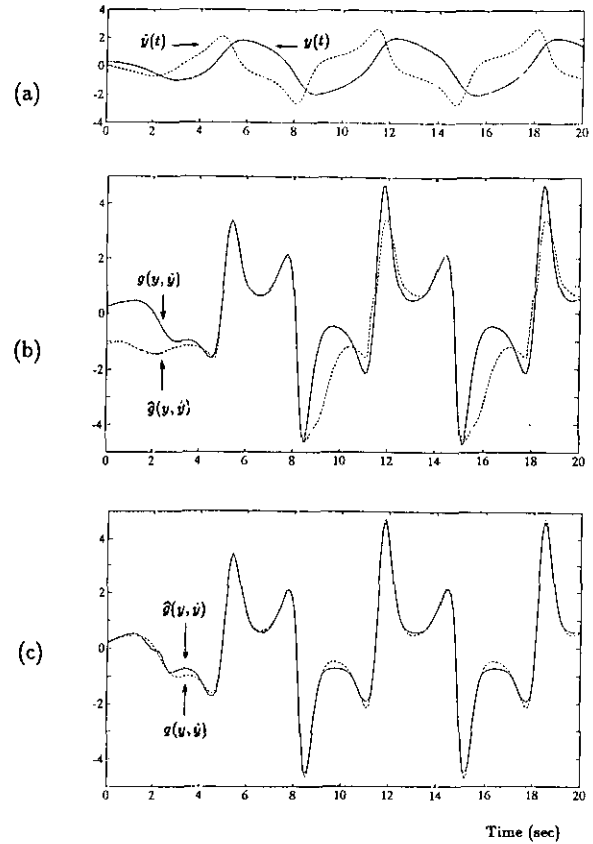


Figure 6. System obeying the Van der Pol equation; (a) neural net inputs $y(t)$, $\dot{y}(t)$; (b) network response when trained over $t = [3.6, 8.5]$ s; (c) network response when trained over $t = [0.0, 14.5]$ s.

- (1) fits a surface in multi-dimensional space to the data set (the 'learning' phase), and
- (2) later on uses the input/output functional relation of the trained network to perform 'interpolation' at points where the training set does not have data points (the 'generalization' phase).

3.2. Identification of systems obeying the Van der Pol equation

In this section, another example of a non-linear single degree-of-freedom system is presented [77]. The restoring function is of the form

$$g(y, \dot{y}) = \varepsilon(y^2 - 1)\dot{y} + y \quad (38)$$

with excitation $u(t) = 0$ and mass $M = 1$. This system corresponds to the homogeneous Van der Pol equation, which is used to model several mechanical and electrical systems. The non-linear damping term $\varepsilon(y^2 - 1)\dot{y}$ gives rise to self-sustained oscillations and limit cycle behaviour.

The training set (inputs: y , \dot{y} , output: g) was obtained from the solution of (38) with $\varepsilon = 1$ and initial condition vector $[y(0), \dot{y}(0)]^T = [0.25, 0.0]^T$, as shown in figure 6(a). A three-layer feedforward network with static nodes is used, trained by the classical back propagation algorithm. The network has two inputs, one output and 15 and 10 nodes in the first and second hidden layers respectively.

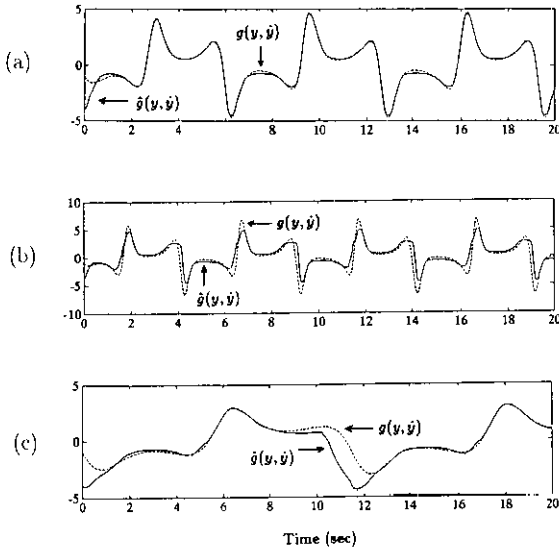


Figure 7. System obeying the Van der Pol equation; (a) network response with initial conditions $[y(t), \dot{y}(t)] = [-1.0, -1.0]^T$; (b) network response with $\varepsilon = 2.0$; (c) network response with $\varepsilon = 0.3$.

This choice was made after experimentation with neural networks of different topologies and sizes.

The effect of several conditions on the network performance was examined. Specifically the following parameters were varied.

(a) *The length of the training set.* Figure 6(b) shows the network response when training was done with data obtained over the time interval $t = [3.6, 8.5]$ s. It is seen that the net performs well only for points within the training interval. However, when the training interval is expanded to $t = [0.0, 14.5]$ s the net is able to follow the system's limit cycling as well as the transition from the initial conditions to the limit cycle (figure 6(c)).

(b) *The initial conditions.* Next, input data $\{y, \dot{y}\}$ were generated by solving (38) with the new initial conditions

$$[y(0), \dot{y}(0)]^T = [-1.0, -1.0]^T.$$

The function $g(y, \dot{y})$ generated with these data is shown in figure 7(a). These pairs $\{y, \dot{y}\}$ were given as inputs to the neural net of part (a) (which was trained over the interval $t = [0.0, 14.5]$ s, with data corresponding to initial conditions $[y(0), \dot{y}(0)]^T = [0.25, 0.0]^T$).

The response $\hat{g}(y, \dot{y})$ of the neural network when fed these new data is also shown in figure 7(a). It is seen that the network tracks the system very well, since it has already been trained and learned the system's limit cycling behaviour (in the training procedure of part (a)).

(c) *The parameter ε .* Changing ε is expected to have a greater effect on the network's prediction ability, since a large ε produces responses y, \dot{y} out of the range of values used during training. In figure 7(b) the graph of $g(y, \dot{y})$ was generated by solving (38) with $\varepsilon = 2.0$ and initial conditions $[y(0), \dot{y}(0)]^T = [-1.0, -1.0]^T$. The solution vectors $\{[y, \dot{y}]\}$, were fed as inputs to the same neural net of part (a) (which was trained over the interval

$t = [0.0, 14.5]$ s, with data corresponding to initial conditions $[y(0), \dot{y}(0)]^T = [0.25, 0.0]^T$ and $\varepsilon = 1.0$).

The response $g(y, \dot{y})$ of the neural net when fed these inputs is also given in figure 7(b). It is seen that the network is now missing the peaks, since it has never been trained with data of such magnitude.

The results obtained when the network is fed inputs generated from (38) with $\varepsilon = 0.3$ and initial conditions $[y(0), \dot{y}(0)]^T = [-1.0, -1.0]^T$, are shown in figure 7(c). The net now gives better results since it has already been exposed to the ranges of y, \dot{y} and $g(y, \dot{y})$ during the training session of part (a) above.

3.3. Identification of a non-linear dynamic system by nets with dynamic neurons

In this section an application of neural networks with dynamic neurons is presented [61, 62]. The system to be identified is a two degree of freedom robotic manipulator, described by the following non-linear vector differential equation

$$\tau(t) = M(\theta(t))\ddot{\theta}(t) + C(\theta(t), \dot{\theta}(t))\dot{\theta}(t) \quad (39)$$

where $\tau(t)$ is a 2×1 vector of joint torques; $\theta(t)$ is a 2×1 vector containing the joint variables (joint angles); $M(\theta(t))$ is a 2×2 matrix, whose elements are non-linear functions of $\theta(t)$, representing the contribution of the inertial forces to the dynamical equation; $C(\theta(t), \dot{\theta}(t))\dot{\theta}(t)$ represents the coriolis forces.

The problem we are dealing with is that of modelling the forward dynamics of this system: given the vector $\tau(t)$ of input torques, reproduce the output vector $\theta(t)$ of joint angles. The training set consists of the time histories of input torques which are sinusoidal signals with time-varying frequencies, ranging from 0.001 rad s^{-1} to 1.0 rad s^{-1} , and the corresponding output joint angles $\theta(t)$. Initially, a static network with only a sigmoid static function $\gamma(x)$ at the nodes was chosen and trained over the set of input/output data. The network chosen was a three-layer net, with two inputs, two outputs and ten nodes at each hidden layer. The results of the network identification are shown in figure 8(a) for the angle $\theta_1(t)$. It is seen that the net does not perform an accurate identification, since no dynamics were included in the network architecture. Next, a network with dynamic neurons is used and trained by the distributed dynamic back propagation algorithm. The network chosen is of the same topology and size as the static neural net (three-layered net, with two inputs, two outputs and ten nodes at each hidden layer). A linear dynamical system of the form $\dot{x} = -ax + bu$ was employed at the nodes in addition to the static sigmoidal non-linearity $\gamma(x)$. The network performs a very good modelling of the non-linear robotic system, as can be seen from figure 8(b), for the variable $\theta_1(t)$. Similar results were obtained for the second joint variable $\theta_2(t)$.

The authors' experience in networks with dynamic neurons provides strong indications that networks with distributed dynamics are a powerful tool for modelling unknown non-linear dynamic systems.

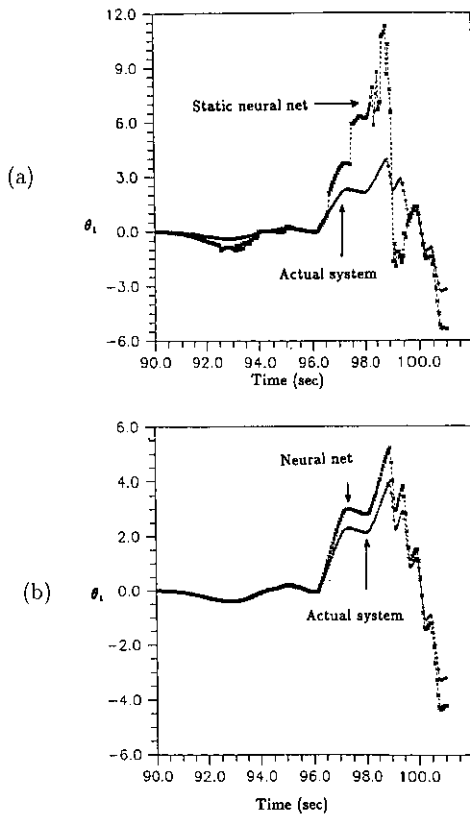


Figure 8. Modelling of a robotic system (a) using a static neural network; (b) using a network with dynamic neurons.

4. Summary and Conclusions

This paper presents the theoretical development of a new method for identifying structure-unknown non-linear dynamic systems typically encountered in the applied mechanics field. The method is based on the use of neural networks as system identifiers. The relevant background is discussed, the identification problem is formulated and an identification procedure which handles a general class of structural systems is developed. The similarity of this approach to a qualitatively analogous non-parametric identification procedure, previously developed by the authors, is also discussed.

Illustrative examples involving the use of multilayer, feedforward neural networks to identify representative non-linear systems are presented.

References

- [1] Jet Propulsion Laboratory, 1988 *Model Determination for Large Space Systems Workshop*, (California Institute of Technology, 22–24 March 1988) Report No. JPL D-5574
- [2] Housner G W and Masri S F (Editors) 1990 *Proceedings of the US National Workshop on Structural Control Research* (University of Southern California, Los Angeles, 25–26 October 1990) USC Publication No CE-9013
- [3] Natke H G and Yao J T P 1988 *Structural Safety Evaluation Based on System Identification Approaches* (Braunschweig: Frieder Vieweg)
- [4] Astrom K J and Eykoff P 1971 System identification—a survey *Automatica* **7** 123–162
- [5] Beck J L 1978 Determining models of structures from earthquake records, Earthquake Engineering Research Laboratory, California Institute of Technology, Pasadena
- [6] Natke H G 1982 *Identification of Vibrating Structures* (Wien, New York: Springer)
- [7] Union College 1991 *Proc. International Modal Analysis Conference*, (Schenectady, NY)
- [8] McClelland W S and Pitts W H 1943 A logical calculus of the ideas immanent in nervous activity *Bulletin of Mathematical Biophysics* **5** 115–133
- [9] Hebb D O 1949 *The Organization of Behaviour* (New York: Wiley)
- [10] Rosenblatt F 1962 *Principles of Neurodynamics: Perceptrons and the theory of Brain Mechanisms* (New York: Spartan Books)
- [11] Widrow B 1962 Generalization and information storage in networks of ADALINE neurons *Self Organizing Systems* (ed G T Yovits) (New York: Spartan Books)
- [12] Minsky M and Papert S 1969 *Perceptrons: an Introduction to Computational Geometry* (MIT Press)
- [13] Werbos P 1974 Beyond regression: new tools for prediction and analysis in the behavioral sciences, *Ph.D. Thesis*, Harvard University
- [14] Albus J S 1975 A new approach to manipulator control: the cerebellar model articulation controller (CMAC) *Trans. ASME, J. Dynam. Syst. Measur. Contr.* **97** 220–227
- [15] Grossberg S 1976 Adaptive pattern recognition and universal recording: part I, parallel development and coding of neural feature detectors *Biol. Cybern.* **23** 121–134
- [16] Hopfield J J 1982 Neural networks and physical systems with emergent collective properties *Proc. Nat. Acad. Sci. USA* **79** 2554–2558
- [17] Hinton G E, Sejnowski T J and Ackley D 1984 *Boltzmann Machines: Constraint Satisfaction Networks that Learn* Tech. Rep. CMU CS-84, 111, Carnegie-Mellon University, Pittsburgh
- [18] Kohonen T 1984 *Self Organization and Associative Memory* (Heidelberg: Springer)
- [19] McLelland J L and Rumelhart D E 1986 *Parallel Distributed Processing* vol 1 (MIT Press)
- [20] Mead C 1989 *Analog VLSI for Neural Systems* (New York: Addison-Wesley)
- [21] Aleksander I and Morton H 1990 *An Introduction to Neural Computing* (London: Chapman and Hall)
- [22] Khanna T 1990 *Foundations of Neural Networks* (Reading, MA: Addison-Wesley)
- [23] Nelson M M and Illingworth W T 1991 *A Practical Guide to Neural Nets* (Reading MA: Addison-Wesley)
- [24] Lippmann R P 1987 An introduction to computing with neural nets *IEEE ASSP Mag.* April, 4–22
- [25] Graupe D 1976 *Identification of Systems* (Huntington, NY: Robert E. Krieger Publishing)
- [26] Greenberg H 1951 A survey of methods for determining stability parameters of an airplane from dynamic flight measurements NACA TN 2340
- [27] Shinbrot M 1951 A least squares curve fitting method with applications to the calculation of stability coefficients from transient response data NACA TN 2341
- [28] Shinbrot M 1952 A description and a comparison of certain nonlinear curve fitting techniques, with applications to the analysis of transient-response data NACA TN 2622
- [29] Briggs B R and Jones A L 1953 Techniques for

- calculating parameters of nonlinear dynamic systems from response data NACA TN 2977
- [30] Shinbrot M 1954 On the analysis of linear and nonlinear dynamical systems from transient-response data NACA TN 3288
- [31] Klotter K 1953 The attenuation of damped free vibrations and the derivation of the damping law from recorded data, Stanford University Division of Engineering Mechanics, Contract N6-ONR-251, Technical Report 23, Nov. 1
- [32] Shinbrot M 1957 On the analysis of linear and nonlinear systems *Trans. ASME* **79** 547-552
- [33] Kohr R H 1963 A method for determination of a differential equation model for simple nonlinear systems *IEEE Trans. Electronic Computers* **12** 394-400
- [34] Hoberock L L, Kohr R H 1967 An experimental determination of differential equations to describe simple nonlinear systems *Trans. ASME, J. Eng.*, 393-398
- [35] Sprague C H, Kohr R H 1969 The use of piecewise continuous expansions in the identification of nonlinear systems *J. Basic. Eng.*, ASME Transactions Series D, vol 91, 173-178
- [36] Sehitoglu H and Klein R E 1975 Identification of multi-valued and memory nonlinearities in dynamic process *Simulation* Sept. 86-92
- [37] Masri S F and Caughey T K 1979 A nonparametric identification technique for nonlinear dynamic problems *J. Appl. Mech., Trans. ASME* **46** 433-447
- [38] Masri S F, Sassi H and Caughey T K 1982 Non-parametric identification of nearly arbitrary nonlinear systems *J. Appl. Mech. Trans. ASME* **11** 1-2
- [39] Shinozuka M, Yun C-B and Imai H 1982 Identification of linear structural dynamic systems *J. Eng. Mech. Division, Trans. ASCE* **108**(1) 1371-1390
- [40] Masri S F, Miller R K, Sassi H and Caughey T K 1984 A method for reducing the order of nonlinear dynamic systems *J. Appl. Mech.* **51** 391-398, *Trans. ASME*, USC Report USC-CE-8307
- [41] Tomlinson G R 1985 *Dynamics of Nonlinear Systems* Report, Dept of Mechanical Engineering, Heriot-Watt University, Edinburgh, UK
- [42] Hac A and Spanos P D 1987 Time domain method for parameter identification of structures *Eng. Mech.—6th Conference Abstracts, Proc. ASCE*, 20-22 May, p 323
- [43] Crawley E F and Aubert A C 1986 Identification of nonlinear structural elements by force-state mapping *AIAA J.* **24** 155-162
- [44] Crawley E F and O'Donnell K J 1986 Identification of nonlinear system parameters in joints using the force-state mapping technique *AIAA J.* **24** 659-667
- [45] Masri S F, Miller R K, Saud A F and Caughey T K 1987 Identification of nonlinear vibrating structures; part I: formulation *Trans. ASME, J. Appl. Mech.* **109** 918-922
- [46] Masri S F, Miller R K, Saud A F and Caughey T K 1987 Identification of nonlinear vibrating structures; part II: applications *Trans. ASME, J. Appl. Mech.* **109** 923-929
- [47] Masri S F, Miller R K, Traina M-I and Caughey T K 1991 Development of bearing friction models from experimental measurements *J. Sound & Vibration* **148** 455-475
- [48] Anderson J C, Masri S F, Miller R K, Sassi H and Caughey T K 1984 Identification of a nonlinear building model from response measurements under earthquake excitation *Proc. Eighth World Conf. on Earthquake Engineering (San Francisco, CA)*
- [49] Masri S F, Miller R K, Sassi H and Caughey T K 1984 Time domain analysis of nonlinear vibration data *Proc. Second Int. Conf. on Recent Advances in Structural Dynamics (Southampton, UK)* USC Report No. USC-CE-8300
- [50] Masri S F, Miller R K, Saud A F and Caughey T K 1985 A time-domain method for the dynamic modeling of jointed structural systems. Presented at the AIAA 26th Structures, Structural Dynamics and Materials Conference (Orlando, FL) 15-17 April. Proc. Paper No. AIAA-85-0709-CP
- [51] Nguyen D H and Widrow B 1990 Neural networks for self learning control systems *IEEE Control Syst. Mag.* **10** 18-23
- [52] Narendra K S and Parthasarathy K 1990 Identification and control of dynamical systems using neural networks *IEEE Trans. Neural Networks* **1** 4-27
- [53] Chu S R, Shoureshi R and Tenorio M 1990 Neural networks for system identification *IEEE Control Syst. Mag.* **10** 31-35
- [54] Kraft L G and Campagna D P 1990 A comparison between CMAC neural network control and two traditional adaptive control systems *IEEE Control Syst. Mag.* **10** 36-43
- [55] Lan, Ming-Shong 1989 Adaptive control of unknown dynamical systems via neural network approach *Proc. Amer. Contr. Conf. (Pittsburgh)* 910-915
- [56] Chen S, Billings S A, Cowan C F N and Grant P M 1990 Non-linear systems identification using radial basis functions *Int. J. Syst. Sci.* **21** 2513-2539
- [57] Poggio T and Girosi F 1990 Networks for approximation and learning *Proc. IEEE* **78** 1481-1497
- [58] Narendra K S and Parthasarathy K 1991 Gradient methods for the optimization of dynamical systems containing neural networks *IEEE Trans. Neural Networks* **2** 252-262
- [59] Werbos P 1990 Backpropagation through time: what it does and how to do it *Proc. IEEE, Special Issue on Neural Networks* **78** October
- [60] Polycarpou M and Ioannou P A 1991 Identification and control of nonlinear systems using neural network models: design and stability analysis *Proc. Workshop on Neural Networks in Robotics (Center for Neural Engineering, University of Southern California, 24-26 October 1991)*
- [61] Chassiakos A G, Kosmatopoulos E B and Christodoulou M A 1991 Identification of robot dynamics by neural networks with dynamic neurons *Proc. Workshop on Neural Networks in Robotics (Center for Neural Engineering, University of Southern California, 24-26 October 1991)*
- [62] Kosmatopoulos E B, Chassiakos A G and Christodoulou M A 1991 Robot identification using neural networks *IEEE 30th Conf. on Decision and Control*
- [63] Cohen M A and Grossberg S 1983 Absolute stability of global pattern formation and parallel memory storage by competitive neural networks *IEEE Trans. on Systems, Man and Cybernetics* **13** 815-826
- [64] Pineda F J 1987 Generalization of back propagation to recurrent networks *Phys. Rev. Lett.* **59** 2229-2232
- [65] Caughey T K 1960 Random excitation of a system with bilinear hysteresis *ASME J. Appl. Mech.* **27** 649-652
- [66] Caughey T K 1975 Nonlinear analysis, synthesis and identification theory *Proc. Symp. on Testing and Identification of Nonlinear Systems (California Institute of Technology)* pp 1-14
- [67] Iwan W D 1965 The steady-state response of the double bilinear hysteretic system *ASME J. Appl. Mech.* **32** 921-925
- [68] Iwan W D 1966 A distributed-element model for hysteresis and its steady-state dynamic response *ASME J. Appl. Mech.* **33** 893-900
- [69] Iwan W D and Lutes L D 1968 Response of the bilinear hysteretic system to stationary random excitation *J. Acoust. Soc. Am.* **43** 545-552

- [70] Lutes L D and Takemiya H 1974 Random vibration damping of yielding oscillator *J. Eng. Mech. Div., Proc. ASCE* **100** 343–358
- [71] Baber T T and Wen Y K 1981 Random vibration of hysteretic degrading systems *J. Eng. Mech.* **107** 1069–1087
- [72] Andronikou A M and Bekey G A 1984 Identification of hysteretic systems *Proc. 18th IEEE Conf. on Decision and Control* pp 1072–1073
- [73] Yar M and Hammond J K 1987 Parameter estimation for hysteretic systems *J. Sound and Vibration* **117** 161–172
- [74] Chassiakos A G and Masri S F 1991 Identification of the internal forces of structural systems using neural networks *IEEE-INNS, Int. Joint Conf. on Neural Networks (Seattle, WA, July 1991)*
- [75] Chassiakos A G and Masri S F 1991 Identification of the internal forces of structural systems by feed-forward multilayer networks *Computing Syst. in Eng.* **2** 125–134
- [76] Masri S F, Chassiakos A G and Caughey T K 1992 Identification of nonlinear dynamic systems using neural networks (to appear in *J. Appl. Mech., Trans. ASME*)
- [77] Chassiakos A G and Masri S F 1991 Neural network based identification of structural systems *Proc. IMACS-IFAC Symp. on Parallel and Distributed Computing in Engineering Systems (Corfu, Greece, 23–28 June 1991)*